An International Journal
computers & mathematics
with applications

PERGAMON Computers and Mathematics with Applications 37 (1999) 41–45

The Estimation of Parameters in Time-Dependent Transport Problems: Dynamic Programming and Associative Memories

R. E. KALABA
Departments of Electrical Engineering, Biomedical Engineering, and Economics
University of Southern California
Los Angeles, CA 90089, U.S.A.

H. H. NATSUYAMA AND S. UENO
Information Science Laboratory
Kyoto School of Computer Science
7 Tanaka monzencho, Sakyo-ku
Kyoto 606, Japan
hnatsu@earthlink.net

(Received April 1998; accepted May 1998)

Abstract—Time-dependent radiation and energy transport problems are important in atmospheric science, medicine, biochemistry, and other areas. To determine external energy fields, direct problems (in which parameters are known) can be solved computationally by numerical integration followed by the numerical inversion of Laplace transforms. On the other hand, this paper treats inverse problems of estimating transport parameters on the basis of external observations of radiant intensity. These problems are approached using associative memory neural networks whose associated least squares problem is solved using a new dynamic programming algorithm. The quality of the estimates in the presence of noise in measurements is studied. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords—Dynamic programming, Transport processes, Parameter estimation.

1. INTRODUCTION

The partial differential equations of time-dependent transport can be solved by numerical integration of differential equations followed by the numerical inversion of Laplace transforms, a technique that is described in [1]. Tables of intensities reflected from slabs after multiple scatterings have been computed and are reported in [1,2]. Inverse problems of estimating internal transport parameters on the basis of reflection measurements have been approached using associative memory neural networks [3,4]. In this paper, the resultant least squares problem is solved using a dynamic programming algorithm that is described in [5].

 Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX
PII: S0898-1221(98)00252-1

2. STATEMENT OF PROBLEM

Consider the time-dependent multiple scattering problem [1] of a homogeneous finite slab of thickness one illuminated by parallel rays of radiation at the top, whose intensity varies in time as the delta-function of zero. The angle of incidence is 13.0 degrees from the normal, as is the angle of reflection of the reflected intensities. The intensity is observed at three or four instants of time, depending on the trial. The intensities, obtained from the tables of [2], are given in Table 1 for albedoes 0.2, 0.6, and 1.0. We pose the inverse problem of estimating the albedo of a slab when given a set of observations from column 2, 3, or 4 of Table 1.

Table 1. Reflected intensities at three and four times of observation for slabs with albedo 0.2, 0.6, and 1.0

	Reflected Intensities		
Time	Albedo 0.2	Albedo 0.6	Albedo 1.0
t1	.0245	.0740	.1194
t2	.0221	.0689	.1033
t3	.0185	.0614	.0695
t4	.0136	.0496	.0188

Associative Memory Approach

Our approach to the estimation problem is that of associative memory neural networks [3]. The associative memory approach to the inverse problem of estimating albedoes has two parts:

(1) given a set of training cases of albedoes and corresponding observations, capture the information in a memory matrix;

(2) using the memory matrix, and being given a set of observations, estimate the corresponding albedo.

Let us formulate this problem in terms of a parameter vector a, which here is of dimension 1×1; a measurement or observation column vector s, which is of dimension $n \times 1$, where n is the number of measurements in a set of observations. Let us suppose that $n = 4$, for the sake of definiteness. Then the relationship between a and s is presumed to have the form given by the "memory vector" M, a row vector with n components:

$$a = Ms. \tag{1}$$

The dimensions of this equation are $(1 \times 1) = (1 \times n)(n \times 1)$. Equation (1) is our estimator equation, to be used when a set of observations is obtained and a parameter vector is to be estimated.

To determine what the vector M should be, we provide information on training cases: for various given albedoes of a medium, define the corresponding observations of intensity. For each pair of albedo and observations we have a set of approximate relations of the form of (1). We combine the a vectors, column by column, into a matrix A, so that A is of dimension $1 \times L$, where L is the number of training sets. Also, we form the matrix S out of the column vectors s, so that S is of dimension $n \times L$. Then we impose the relation,

$$A = MS, \tag{2}$$

in a least squares sense, on the matrix M.

Least Squares Problem

We introduce the following vectors and matrices, (denoting the transpose of a vector or a matrix by the prime symbol):

$$b = a', \tag{3}$$

$$A = S', \tag{4}$$

$$x = M'. \tag{5}$$

Then the problem takes the form,

$$Ax = b, \tag{6}$$

to be solved in the least squares sense. By also requiring the minimum norm of x, we have the problem stated in the form treated in [5].

Dynamic Programming Solution Method

The least squares minimum norm problem is solved by the dynamic programming algorithm. The columns of A are the a_i, $i = 1, \dots, n$ (the number of observations) and the number of rows is the number of training cases, L. In other words, each row of A is an observation vector. The column vector b has the L components which are the given parameters, in this case, the albedoes. The solution column vector x gives the desired memory row vector, M, which has n components.

The algorithm is described by the following procedure.

1. Input the A matrix and the b vector.
2. Sweep forward from columns 1 through n and store n pairs of auxiliary vectors using recurrence relations with initial conditions.
3. Sweep backward and determine the components of the vector x from the n^{th} component to the first.

The details for the dynamic programming algorithm are found in [5]. From the M-vector and the given observations, we get our parameter estimate using (1).

3. COMPUTATIONAL METHOD

We implement the dynamic programming algorithm and the parameter estimation of (1) in a Matlab program. We perform a series of tests before the program is accepted. For example, we use accurate values in the training sets to compute the memory vector; then the memory vector is used on an accurate set of observations to recover the albedo. With accurate observations, accurate estimates of albedo are made. Then we carry out computational experiments.

4. COMPUTATIONAL RESULTS

4.1. Noise-Free Training

Noise-free data in training sets

We determine the memory vector using these three observations per set, and we repeat the memory vector computation using four observations. The memory matrices computed for three and four observations are presented below:

$$M = [6.2454 \quad 3.8816 \quad -2.1101],$$

and

$$M = [7.2252 \quad 2.3038 \quad -1.4712 \quad -0.1065].$$

Observe that the heaviest weight is placed on the first observation. Examination of the computations reveals that the observations may be linearly dependent, or nearly so.

Noisy data in estimation experiments

Experiments to estimate albedo are made with noisy observations. A noise vector whose components are random numbers obtained from a distribution that is uniform in the interval $(-1, +1)$ and multiplied by 0.01 is added to the accurate observations to simulate a noisy observation vector. The noisy observations are listed in Table 2. Observe that the noise causes the previously monotonically decreasing observations to fluctuate. Estimates of albedo, using noisy observations, are shown below with three and four observations, respectively.

Albedo estimates:	0.1746,	0.5749,	0.9748,	(3 observations)
	0.2083,	0.6083,	1.0083,	(4 observations)

We find that even with fairly large amounts of noise, quite satisfactory estimates are obtained.

Table 2. Noisy measurements of reflected intensities at three and four times of observation for slabs with albedo 0.2, 0.6, and 1.0

	Noisy Measurements of Reflected Intensity		
Time	Albedo 0.2	Albedo 0.6	Albedo 1.0
t1	.0216	.0711	.1165
t2	.0284	.0752	.1096
t3	.0087	.0516	.0597
t4	.0064	.0424	.0116

4.2. Noisy Training

Training using noisy observations and estimation using noisy observations

When we train using noisy observations from Table 2, the memory vector obtained for three observations per set is changed dramatically to:

$$M = [\,1.2842 \quad 10.8817 \quad -4.0287\,],$$

Highly accurate estimates of albedo–0.2, 0.6, and 1.0, respectively, to four decimal places–are obtained when using observation vectors that are the same as those used in training. In the following experiments, we use different sets of noisy observation vectors for estimation. The results are as follows. When the amount of noise added is 0.001 times a random number, the estimates are:

Albedo estimates: 0.2019, 0.6019, 1.0019.

When the noise is 0.004, the estimates are degraded to:

Albedo estimates: 0.1459, 0.5459, 0.9459.

Clearly, noise in measurements plays an important role in estimation. It would be desirable to conduct a Monte Carlo study of the effect of noise in observations on estimates of parameters.

4.3. Effective Training

Training with small amounts of noise

It seems prudent to train with small amounts of noise, such as 0.001 times a random number. This is done and the results are as follows. The noise improves the conditioning of the problem, so that the columns are not so linearly dependent (i.e., their inner products are greater than zero). With a tolerance of 10^{-6} for the dependency test, the third column may be considered independent. The memory vector thus obtained,

$$M = [9.2983 \quad -0.4210 \quad -0.9455],$$

places great weight on the first observation, as may be expected. Estimates using observation vectors containing 0.002 times a random number are excellent:

Albedo estimates: 0.1966, 0.5966, 0.9966.

4.4. Summary of Numerical Experiments

When observations are not independent, there may be difficulty in obtaining reliable estimates. The dynamic programming approach picks out situations in which this is happening and alerts the analyst to this fact. As a practical rule of thumb for conducting controlled experiments, the addition of a small amount of noise into the training data will help the conditioning of the problem, and thus, improve the estimation.

5. DISCUSSION

We have found that dynamic programming is an effective way to solve this class of least squares inverse problems. Further studies are planned in order to investigate the robustness of this approach in the face of different types of errors. By conducting a Monte Carlo study of the effect of noise in observations on estimates, it may be possible to develop confidence figures for this estimator. Other investigations will include the extension of this approach to multiparameter estimation.

REFERENCES

1. R. Bellman, H. Kagiwada, R. Kalaba and M. Prestrud, *Invariant Imbedding and Time-Dependent Transport Processes*, Elsevier, (1964).
2. R. Bellman, H. Kagiwada and R. Kalaba, *Time-Dependent Diffuse Reflection from Slabs with Multiple Scattering*, RM-5070-PR, The Rand Corporation, (1966).
3. H. Kagiwada, R. Kalaba, S. Timko and S. Ueno, Associative memories for system identification: Inverse problems in remote sensing, *Mathl. Comput. Modelling* **14**, 200-202, (1990).
4. H.H. Natsuyama, S. Ueno and A.P. Wang, *Terrestrial Radiative Transfer: Modelling, Computation, and Data Analysis*, Springer-Verlag, (1998).
5. R. Kalaba and H. Natsuyama, Dynamic programming and minimal norm solutions of least squares problems, University of Southern California, (April 1998).